Transcrição

Continuando com o assunto de Distribuição de Frequências, trabalharemos com **variáveis quantitativas** neste passo, a

O primeiro passo é gerarmos uma maneira de categorizá-las para construirmos a Distribuição, e nesta etapa aprenderem

Em nosso notebook, na parte "Passo 1 - Especificar os limites de cada classe" que diz respeito ao assunto, encontrarem

Esta classificação se origina de um trabalho que faz a divisão em quantidade de **salários mínimos** que compõe o rendi

- **A**: acima de 20;
- **B**: de 10 a 20;
- **C**: de 4 a 10;
- **D**: de 2 a 4;
- **E**: de zero até 2 salários mínimos.

Como a pesquisa PNAD na qual baseamos nossos dados foi realizada em 2015, o valor do salário mínimo era de **R$78**

Com base nisso, faremos o cálculo dos valores em reais de cada classificação apresentada.

- **A**: acima de R$15.760,00;
- **B**: de R$7.880,00 até R$15.760,00;
- **C**: de R$3.152,00 até R$ 7.880,00;
- **D**: de R$1.576,00 até R$3.152,00;
- **E**: de zero até R$1.576,00.

Primeiro, descobriremos quais são os **valores mínimos** e **máximos** da variável para construirmos a distribuição.

Na célula, usaremos os `dados` de `.Renda` com `.min()`.

```
dados.Renda.min()
```

O retorno será `0` para o valor mínimo, logicamente. Depois, aplicaremos a mesma linha com `.max()` para o máximo

```
dados.Renda.max()
```

Como saída, o sistema apresentará o valor `200000` relativo ao valor máximo da variável.

Estas respostas nos farão refletir se estamos recebendo valores corretos ou se é um **outlier**. De qualquer maneira, anota

Continuando, usaremos uma funcionalidade do Pandas chamada **`cut()`**, a qual precisa de alguns parâmetros que serã

Começaremos pelos **limites** das classes de renda, e os colocaremos dentro da variável `classes`, a qual será uma lista

O segundo limite `1576` será extraído da classe "E", o terceiro `3152` da "D", o quarto `7880` da "C", o quinto `15760`

```
classes = [0, 1576, 3152, 7880, 15760, 200000]
```

Executada a célula e criada a variável `classes`, faremos uma visualização mais clara com labels para as categorias.

Na linha seguinte, criaremos a nova variável `labels` recebendo uma lista Python de novo, contendo os nomes entre a maior valor `'A'`.

```
labels = ['E', 'D', 'C', 'B', 'A']
```

Feito isso, entenderemos como o método `cut()` funciona em Pandas. Em "Passo 2 - Criar a tabela de frequências" do

Na primeira célula desta parte, escreveremos `pd.cut()` recebendo a variável `dados.Renda` que estamos trabalhando construí-las.

Como já criamos labels e queremos exibi-las, passaremos sua variável como terceiro parâmetro também chamado `lab`

Por *default*, este método não inclui a classe inferior `0` nas classes, então precisaremos indicar ao `cut()` que queremo

Para isso, usaremos o quarto parâmetro `include_lowest` sendo igual a `True`.

```
pd.cut(x = dados.Renda,
       bins = classes,
       labels = labels,
       include_lowest = True)
```

0	E
1	E
2	E
3	C
4	E

Executando esta célula, o sistema criará uma *series* com um índice igual ao `DataFrame` onde a variável dos registros

Se clicarmos no botão "CODE" que aparece na própria célula que estamos escrevendo, criaremos uma nova anterior à

Nesta nova célula, plotaremos os `dados` com somente os cinco primeiros valores utilizando `head()`, e executaremos

```
dados.head()
```

	UF	Sexo	Idade	Cor	Anos de Estudo	Renda	Altura
0	11	0	23	8	12	800	1.603808
1	11	1	23	2	12	1150	1.739790
2	11	1	35	8	15	880	1.760444
3	11	0	46	2	6	3500	1.783158
4	11	1	47	8	9	150	1.690631

```
pd.cut(x = dados.Renda,
       bins = classes,
       labels = labels,
       include_lowest = True)
```

0	E
1	E
2	E
3	C
4	E

Com estes dados, analisaremos os registros para vermos se realmente os valores de `Renda` correspondem à classificaç

Poderemos excluir a linha de `dados.head()` para não nos confundirmos, acessando o menu da própria célula e clican

Continuando com esses dados, seguiremos a mesma metodologia adotada anteriormente; passaremos para o método `v` função em questão para fazermos a contagem.

Agora, poderemos chamá-lo por meio do próprio Pandas, escrevendo apenas `pd.value_counts()` recebendo todo o c

```
pd.value_counts(
    pd.cut(x = dados.Renda,
           bins = classes,
           labels = labels,
           include_lowest = True)
)
```

E	49755
D	16700
C	7599
B	2178
A	608

```
Name: Renda, dtype: int64
```

Nesta execução, veremos a contagem feita da maneira como queríamos.

Este comando é o mesmo que fizemos anteriormente com `frequencia`. Logo, poderemos chamar esta variável para a

```
frequencia = pd.value_counts(
    pd.cut(x = dados.Renda,
               bins = classes,
               labels = labels,
               include_lowest = True)
)
frequencia
```

```
E    49755
D    16700
C     7599
B     2178
A      608
Name: Renda, dtype: int64
```

Na célula seguinte, faremos a coluna de `percentual`. Para isso, passaremos o parâmetro `normalize` sendo igual a `T`

```
percentual = pd.value_counts(
    pd.cut(x = dados.Renda,
               bins = classes,
               labels = labels,
               include_lowest = True),
    normalize = True
)
percentual
```

```
E    0.647514
D    0.217335
C    0.098894
B    0.028345
A    0.007913
Name: Renda, dtype: float64
```

Por fim, aplicaremos a mesma técnica feita com `dist_freq_qualitativas`. Inclusive, poderemos copiar e colar esta l

Logo, criaremos a nova variável `dist_freq_quantitativas_personalizadas` sendo igual a `pd.DataFrame()` recebe

Em seguida, mostraremos o nosso resultado chamando a variável e executando a célula.

```
dist_freq_quantitativas_personalizadas = pd.DataFrame(
    {'Frequência': frequencia, 'Porcentagem (%)': percentual}
)
dist_freq_quantitativas_personalizadas
```

	Frequência	Porcentagem (%)
E	49755	0.647514
D	16700	0.217335
C	7599	0.028345
B	2178	0.028345
A	608	0.007913

Porém, a ordenação está sendo feita de `E` até `A`, e queremos seguir a ordem alfabética de `A` até `E`.

Então chamaremos a `dist_freq_quantitativas_personalizadas` com `.sort_index()` para ordenarmos o índice co

```
dist_freq_quantitativas_personalizadas.sort_index(ascending = False)
```

	Frequência	Porcentagem (%)
A	608	0.007913
B	2178	0.028345
C	7599	0.028345
D	16700	0.217335
E	49755	0.647514

Será esta construção da Distribuição de Frequências que analisaremos mais adiante para tirarmos as conclusões.

Observando os dados, é possível visualizarmos a grande **desigualdade social** e precária distribuição de renda exposta
enquanto as maiores rendas estão concentradas em uma porcentagem muito pequena de domicílios que corresponde à

Também veremos estes resultados em forma de gráfico para podermos abordar a **simetria**.

A seguir, continuaremos com este assunto, mas aprenderemos uma outra maneira de criarmos a categorização de variá